



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/788,801

02/27/2004

Scott Musson

BEAS-01382US0

7864

23910 7590 04/14/2008

FLIESLER MEYER LLP
650 CALIFORNIA STREET
14TH FLOOR
SAN FRANCISCO, CA 94108

EXAMINER

HASSAN, RASHEDUL

ART UNIT

PAPER NUMBER

2179

MAIL DATE

DELIVERY MODE

04/14/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/788,801	Applicant(s) MUSSON ET AL.	
	Examiner RASHEDUL HASSAN	Art Unit 2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 24 January 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-16,18-30,32-44 and 46-49 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,3-16,18-30,32-44 and 46-49 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|----------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>10/5/2007, 12/19/2007, 1/25/2008</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Acknowledgement

The Examiner acknowledges and appreciates the Amendments filed on 1/24/2008.

Independent claims 1, 16, and 30 have been amended.

Claims 2, 17, 31 and 45 have been cancelled.

Claims 46-49 have been added as new claims.

Claims 1, 3-16, 18-30, 32-44 and 46-49 are currently pending.

Response to Arguments

Previous objection to the Specification is hereby withdrawn in consideration to the amendment made to claim 30 to recite "machine readable storage medium" which has adequate support in the Specification (see Specification, paragraph [0115] and [0116]).

Previous objection to claim 30 is hereby withdrawn in consideration to the amendment made to claim 30 to end with a "period".

Previous objection to claim 45 is moot since the claim has been cancelled.

Previous rejection to claim 45 under 35 U.S.C. 101 for being directed to non-statutory subject matter is also moot since the claim has been cancelled.

Regarding the previous prior art rejections, Applicants' arguments filed on 1/24/2008 have been thoroughly considered but are not persuasive.

Applicants argued, *"a control tree is one specific way to represent the GUI or alternatively the entire relevant back end processing components/objects implementing the requested GUI as disclosed in the specification. Therefore, a control tree is not and cannot be anticipated by the GUI it represented, because the representation of one or a group of objects is fundamentally different from the object or objects being represented"*. The Examiner disagrees. A "control tree" is not necessarily a separate entity different from the object or objects being represented but only a conceptual visualization of those objects together, considering the hierarchical dependencies and interrelationships between the objects. As such the interrelationships of objects for implanting a GUI together with those objects represents a control tree (see Anuff, Fig. 4).

Applicants further argued, "Similarly, the servers 12a-12n in Anuff et al. are the creator or implementer of GUI not the 'factory' to generate control tree, which is a representation of GUI". The argument is moot considering the interpretation of the term "control tree" discussed hereinabove, since the servers 12a-12n in Anuff are the creator or implementer of the relevant back end processing components/objects implementing the GUI and their hierarchical interrelationship, i.e., servers 12a-12n are the creator (i.e., factory) to generate control tree.

Applicants further argued, "In addition, a life stage that involves a control or control tree such as 'init' that allows a control to perform initialization is also fundamentally different from a constructor in OOP that instantiates a real object". Applicants' argument amounts to a mere allegation without a showing as to why it is

fundamentally different. The Examiner thus maintains the reasoning for the rejection of claim 1 as presented in the previous Office Action.

Applicants have not provided any additional arguments for the claims. Thus the Examiner maintains the prior art rejections for claims 1, 3-16, 18-30, 32-44 provided in the previous Office Action for the reasoning discussed hereinabove, and further rejects the new claims 46-49 as discussed hereinafter.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1, 3-16, 18-30, 32-44 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Independent claims 1, 16, and 30 recite the limitation "***generating the control tree***" (emphasis added). There is insufficient antecedent basis for this limitation in the claims. All dependent claims inherit the indefiniteness from their respective independent claims.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1, 3-13, 15-16, 18-27, 29-30, 32-42, 44 and 46-49 are rejected under 35

U.S.C. 102(b) as being anticipated by Anuff et al. (US 6,327,628 B1) hereinafter Anuff.

For claims 1, 16, and 30, Anuff anticipates a computer implemented method for responding to a request (an example of a request is disclosed in [0039] of the instant specification to be “an HTML request originating from a web browser. Anuff teaches this limitation. See c3:13-22, c6:57-65, also c14:32-36), comprising:

accepting the request (e.g., accepting a request from a browser in a client device 10 sent to a server device 12. See Fig. 1, c3:1-24);

generating the control tree from a factory based on the request (the “control tree” is interpreted to mean the entire relevant back end processing components/objects implementing the requested GUI together with their interrelationships with each other as illustrated in Fig. 4 in Anuff. Also see the detailed explanation hereinafter in this rejection and the “Response to Arguments” section hereinabove. A “factory” can be interpreted as any of the servers 12a-12n taught by Anuff that are said to generate the relevant back end processing components/objects implementing the requested GUI together with their interrelationships with each other, see Anuff: Fig. 1, c3:58-65);

mapping the request to a control tree wherein the control tree is a logical representation of a graphical user interface (GUI) and wherein the control tree includes a set of controls which are related hierarchically to one another including at least one portlet control that represents at least one portlet (A "control tree" is recited in the claim to be "a logical representation of a graphical user interface" containing a set of hierarchically related controls. According to the disclosure, "controls" represent "corresponding graphical and functional elements in web applications ... In one embodiment, a control can be implemented as one or more classes in an object oriented programming paradigm" [0028]. A "logical representation" is nothing more than an abstract idea for conceptually viewing the GUI as a set of related back end processing components. These components can be, but not necessarily, implemented as objects instantiated from classes in an object-oriented programming paradigm (hereinafter referred to as OOP). Therefore, a "control tree" can reasonably be interpreted to mean the entire relevant back end processing components/objects implementing the requested GUI together with their interrelationships with each other (see instant specification [0036]). Similarly, "mapping the request to a control tree" can reasonably be interpreted to mean, identifying the entire relevant back end processing components/objects implementing the requested GUI. Anuff inherently teaches identifying the entire relevant back end processing components/objects implementing the requested GUI. He further teaches, with regard to Fig. 4, that these back end controls/objects are related hierarchically to one another, e.g., A owns B and A is a subclass of B. Thus Anuff teaches mapping the request to a control tree wherein the control tree includes a set of controls which are related hierarchically to one another. He also teaches that the control tree includes at least one portlet control that represents at least one portlet (e.g., Module 29 in Fig. 4, see c6:21-32, and modules 26 in Fig. 2. Additionally see the "Response to Arguments" section hereinabove);

advancing the control tree through at least one lifecycle stage based on the request, wherein the set of controls in the control tree operates to interact with each other and produce response based on the request in the at least one lifecycle stage (For

a control, the lifecycle is defined in the instant disclosure, by a set of methods representing stages in the lifecycle. Life cycle stages are illustrated in Table 3 and appear to be nothing more than various stages of an object, instantiated from a class in the context of OOP, during runtime. Therefore, Anuff's controls for generating a portal GUI, implemented using objects in OOP, inherently advances the objects implementing the GUI through at least one lifecycle stage, e.g., at least the "Init" stage that allows a control to perform initialization based on interaction with each other in order to produce the response, i.e., the GUI, based on the request);

providing the request to a portlet container that contains the at least one portlet (the instant disclosure mentions "In a framework, controls can also serve as containers for other controls. By way of a non-limiting example, a page may contain a booklet and a portlet ..." [0028]. The instant disclosure presents Fig. 2 as an illustration of a control taxonomy in accordance to an embodiment, wherein control/object like a web application 200, portal 202, page 218 etc. containing a portlet object functions as a "portlet container". Anuff teaches server processes 12a-12n that serve as portlet containers. See Fig. 1, c3:58-65); and

aggregating the output of each of the at least one portlets and providing the output to the GUI (in this context, "providing the output to the GUI", is interpreted to mean rendering the output on the display device. Anuff clearly teaches this limitation as shown in Fig. 2).

For claim 46, Anuff teaches a computer implemented method for rendering a graphical user interface (GUI) (see the GUI in Fig. 2), comprising

accepting a request (e.g., accepting a request from a browser in a client device 10 sent to a server device 12. See Fig. 1, c3:1-24);

mapping request to a control tree factory (e.g., mapping the request for content from a browser in a client device 10 to the appropriate server 12 hosting the content. See Fig. 1, c3:58-65);

generating a control tree from the factory (e.g., generating the entire relevant back end processing components/objects implementing the requested GUI together with their interrelationships with each other at the server);

providing a response (e.g., GUI contents as illustrated in Fig. 2 are provided by the servers 12 as response to the request for content from browser in client device 10),

wherein the control tree represent a particular instance of a control taxonomy (a “taxonomy” is an orderly categorization of elements according to their relationships. Thus the entire relevant back end processing components/objects implementing the requested GUI represent a particular instance of a control taxonomy, such as one illustrated in Fig. 4) and a control within the control tree operates to process the request, interact with each other and produce a response (e.g., Anuff teaches throughout the reference how various controls operate with each other in order to process the request and provide the response for the request. See sections such as 3.1 Components, 3.2 Managers and Services, 3.3 Modules etc.).

For claim 3 and 32, Anuff further anticipates generating a response wherein the response can be used to render at least a portion of the GUI (since the response from servers 12a-12n are used to display modules 26 in portal front page. These modules are objects that encapsulate a specific, bounded portion of the GUI, and allow that portion to be administered as a unit. For example, a module might display news, sports scores, stock quotes, or weather forecasts, c3:2-24 and c6:22-31).

For claim 4, 18 and 33, Anuff further anticipates that the step of generating a control tree from the factory comprises: creating a metadata representation (regarding a “metadata representation” the instant disclosure says, “In one embodiment, the metadata representation can be an XML document or Java class file defined by a schema”) of a control tree; and generating a class to construct the control tree based on the metadata representation (Anuff: c6:34-46).

For claim 5, 19 and 34, Anuff further anticipates that the request is a hypertext transfer protocol request (HTTP) (c6:57-58) and the request originates from a web browser (16 in Fig. 1).

For claim 6, 20 and 35, Anuff further anticipates providing the response to a web browser (Fig. 1, Fig. 2, c13:53-55).

For claim 7, 21, 36, and 47, Anuff further anticipates that the control tree is advanced through the at least one lifecycle stage by an interchangeable lifecycle component (regarding an “interchangeable lifecycle component” the disclosure mentions, in regard to Fig. 8, “The control container can use an interchangeable lifecycle driver 804 to drive the control tree through a sequence of states so that the request can be processed. As with the interchangeable persistence driver, an interface is provided to isolate lifecycle driver implementation details from the control container. This allows for different lifecycle implementations to be interchanged as needed”. As for

what constitutes the “interchangeable lifecycle driver/component”, a reasonable interpretation would be, in absence of any explicit definition of the term in the disclosure and without importing limitations from the disclosure into the claim, to be objects/processes arbitrarily combined or divided into separate software, firmware or hardware components responsible to instantiate and carry out the run-time processing of the relevant back end processing components/objects implementing the requested GUI which is inherent in Anuff).

For claim 8, 22 and 37, Anuff further anticipates that each one of the set of controls can have an interchangeable persistence mechanism (regarding an “interchangeable persistence mechanism” the instant disclosure mentions, in regard to Fig. 8, “Controls in the control tree can make use of a persistence interface that acts as a front-end to an interchangeable persistence driver 806. The persistence interface hides persistence implementation details from controls and allows for a flexible architecture where different persistence providers can be “plugged in” as needed” [0056]. Disclosure also mentions, “Controls have the ability to persist state across HTTP (Hypertext Transfer Protocol) requests. A state management API can be provided to give each control in the tree the ability to persist itself before rendering an HTTP response. When an HTTP submit to the same page is received, this saved state can be used to re-hydrate or restore the control tree from its persisted state. Thus, the same state can be maintained across different instances of the same control tree with minimal effort to the control author. Controls can be persisted using a state management persistence mechanism” [0057]. Anuff teaches object persistence using suitable database interface. See c4:16:32 and c5:45-48)

For claim 9, 23 and 38, Anuff further anticipates that each one of the set of controls can render itself according to a theme (c8: 22-49).

For claim 10, 24 and 39, Anuff further anticipates that each one of the set of controls can interact with another one of the set of controls (c4: 60-61).

For claim 11, 25 and 40, Anuff further anticipates that one of the set of controls can advance through the series of at least one lifecycle stage in parallel with another of the controls (since in OOP, objects can be instantiated in parallel and individually carry on their run-time processing in parallel with another object. Anuff also teaches multithreaded module preparation, c14:31-41).

For claim 12, 26 and 41, Anuff further teaches that a lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose (implicitly taught since objects apparently follow these stages in OOP which is well known to a person of ordinary skill in the art).

For claim 13, 27 and 42, Anuff further anticipates that the response is an hypertext transfer protocol (HTTP) response (c6:61-65).

For claim 15, 29 and 44, Anuff further anticipates that each one of the set of controls can be one of: Book, Page (c4:65), Window, Menu, Layout (c4:66), Portlet (modules, c4:65), Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

For claim 48, Anuff further teaches a "wire-up" service is used in the control tree factory that cause the control tree factory to return a root of a control tree (e.g., a network connectivity component of a server 12 can be interpreted as a "wire-up" service in the server 12 which is necessarily used to provide the root, i.e., the topmost building block of a requested portal page which could be the portal front page).

For claim 49, Anuff further teaches associating a context with a root of the control tree (e.g., Fig. 4 illustrates associating a "PortalPageContext" which is associated with the Portal Front page).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claim 14, 28 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Anuff.

For claim 14, 28, and 43, Anuff does not explicitly teach that controls can raise events and respond to events. However, he explicitly teaches that an object model comprises a collection of objects that work together in documented relationships. Official notice is taken that in object oriented programming communication/co-operation between objects using events was well known in the art at the time of the invention. Therefore, if not already implicitly taught by Anuff, it would have been obvious to a person of ordinary skill in the art to modify his invention so that controls can raise events and respond to events. The motivation for such modification would have been necessitated by the very nature of the GUI (portal) which is an interactive application

and it is well known to a person of ordinary skill in the art that such applications are well suited for an event-driven implementation.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to RASHEDUL HASSAN whose telephone number is (571)272-9481. The examiner can normally be reached on M-F 7:30AM - 4PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2179

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Rashedul Hassan/
Examiner, Art Unit 2179

/Weilun Lo/
Supervisory Patent Examiner, Art Unit 2179